

AMMS-Kurzgefasst

COLLABORATORS

	<i>TITLE :</i> AMMS-Kurzgefasst		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		November 2, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AMMS-Kurzgefasst	1
1.1	AMMS-Kurzgefasst	1

Chapter 1

AMMS-Kurzgefasst

1.1 AMMS-Kurzgefasst

AMMS-Kurzgefasst

AMMS (Amiga Multiuser Mailbox System) ist ein Multiuser-Mailbox-System, worin gleichzeitig mehrere User arbeiten koennen. Dabei wird nicht, wie bei anderen Systemen ueblich, das Programm mehrfach in den Hauptspeicher eingeladen, sondern ein Hauptprogramm (MB) wird am Anfang gestartet. Die einzelnen Ports werden ueber den Befehl PORT hochgefahren. Dabei koennen die Ports in beliebiger Reihenfolge hoch- bzw. runtergefahren werden. Ports koennen Consolen, serielle, Arexx-, Link- oder andere Schnittstellen sein. Momentan existieren nur Ports fuer Batch-Ports, xprlib-Consolen (z.B. VT102) und serielle Schnittstellen, die in fast beliebiger Anzahl gestartet werden koennen. Die Anzahl ist momentan auf 30 Ports begrenzt, eine Erhoehung kann bei Bedarf eingebaut werden.

Grundsaeztliche Daten ueber das AMMS :

Das AMMS basiert auf Libraries, die fast alle Funktionen enthalten. Diese Libraries koordinieren alle Multiuser-Funktionen und stellen ueber 500 Funktionen zur Verfuegung. Diese Libraries haben die Endung .mblib und stehen im Verzeichnis amms/libs.

Das Hauptprogramm MB laedt beim Start alle staendig im Speicher stehenden Daten bzw. alle residenten Befehle nach. Es bietet eine Oberflaeche, ueber die alle Ports kontrolliert werden koennen.

Die Ports sind die eigentlichen Schnittstellen zwischen User und System. Sie koennen einzeln hoch- bzw. runtergefahren werden. Ebenso koennen ihre Settings mit Hilfe des Befehls EDIT PORT modifiziert werden.

Beim Einloggen in einem Port wird die Batchdatei MBBATCH:LOGIN, die fuer die Einlog-Sequenz sorgt, unter dem User LOGIN gestartet. Diese Datei ist beliebig aenderbar und muss den Befehl LOGIN enthalten, ueber den sich die User einloggen. Endet die Datei unter dem User LOGIN, so legt der Port automatisch auf.

Alle User besitzen ein User-Directory (MBUDIR:<USERNAME>/), worin persoenliche Daten gespeichert werden. Unter anderem steht dort die Batchdatei LOGIN, wo sich der User seine eigene Einlog-Sequenz definieren kann. Diese Datei wird aus der Einlog-Batchdatei MBBATCH:LOGIN mit Hilfe des Befehl EXECUTE BATCH gestartet. In dieser Datei steht nun der Aufruf zum Menu oder andere Befehle, die beim Einloggen userspezifisch ausgefuehrt werden sollen.

Neue User stellen ueber den Befehl APPLICATION einen Antrag. Dieser Antrag fuehrt zum sofortigen Eintrag des Users im System. Dabei erhaelt der neue User fast alle Zugriffe und Einstellungen des Systemusers NEWUSER. Ebenso wird das komplette User-Directory des Users NEWUSER dem neuen User als Vorgabe kopiert. Damit steht den Sysop die Moeglichkeit frei, den neuen Usern auch die Login-Datei vorzudefinieren (EDIT BATCH NEWUSER). Der neue User kann sich sofort nach dem Antrag unter seinem Namen im System einloggen.

Wird ein User ueber den Befehl EDIT USER eingetragen, so werden keine Daten als Voreinstellung uebernommen. Diese muss man expliziet mit Hilfe des Befehls von einem anderen User kopieren oder per Hand einstellen. Ebenso wird nur das User-Directory ohne Inhalt erzeugt (bis auf das Directory pmsgs). Daher ist der Eintrag von Usern ueber den Befehl EDIT USER abzuraten.

Alle Befehle, die der Benutzer in der Befehlsebene oder in Batchdateien nutzen kann, sind nicht fest im Programm implementiert, sondern werden von Disk nachgeladen. Damit ergibt sich eine Modularisierung, die ein Erweitern des Systems schnell und einfach ermoeגlicht.

Es gibt zwei Arten von Befehlen, die ausfuehrbaren Befehle (mbcom:, mbres:) und die Batchbefehle (mbbatch:).

Um die Geschwindigkeit des Systems zu erhoeen, sind einige ausfuehrbare Befehle (mbres:) resident. Sie werden beim Starten des Systems nachgeladen und stehen staendig im Speicher.

Batchbefehle koennen ebenfalls resident sein und werden dann ebenso beim Starten des Systems eingeladen.

- Ausfuehrbare Befehle sind in Assembler geschrieben und benutzen ausschliesslich die Funktionen der AMMS-Libraries. Zusaetzlich koennen Programme, die in einer Hochsprache geschrieben sind, aufgerufen werden. Diese koennen allerdings nur ueber eine Interface-Routine alle Library-Funktionen des AMMS benutzen, weil die Library-Funktionen fuer Assemblerprogramme ausgelegt sind.
- Batchbefehle sind ASCII-Files, worin alle Befehle des Systems aufgerufen werden koennen (damit stellt das AMMS einen Interpreter zur Verfuegung). In Batchbefehlen koennen Variablen deklariert und Labels definiert werden, Sprungbefehle und Abfragen ermoeגlichen einen Ablauf von allen Algorithmen. Die Login-Batchdatei MBBATCH:LOGIN stellt eigentlich nichts anderes als ein systeminternen Batchbefehl dar, der automatisch beim Connect gestartet wird.

Das AMMS besitzt grundsaeztlich nur eine interaktive Befehlsobeflaeche. Aber durch die Batchbefehle ist es moeglich, ein menueorientiertes System als komfortable Applikation ueber diese Befehlsobeflaeche zu legen. Dabei sind Menues mit Hotkeys und anderen Spielereien kein Problem.

Alle Verwaltungsaufgaben koennen nur ueber Ports ausfuehrt werden. Somit kann das System sowohl vollstaendig vor Ort als auch aus der Ferne gewartet werden, was gerade bei groesseren Boxen sehr wichtig ist,

weil ein Sysop nicht mehr in der Lage ist, alle Aufgaben zu uebernehmen. Die Verwaltungsbefehle sind alle sehr komfortabel geschrieben.

Sysops mit dem Level 9999 haben grundsatzlich auf alles Zugriff. Ihnen koennen keine Befehle oder Daten vorenthalten bleiben. Brettverwalter muessen keine Sysops sein, sie brauchen nur Zugriff auf die Befehle, die mit der Brettverwaltung zu tun haben. Somit kann jeder User Brettverwalter werden, ohne ein Risiko fuer das System darzustellen.

Das AMMS bietet die portuebergreifende Kommunikation. Man kann einen anderen User mit Hilfe des Befehls MONITOR zuschauen und Eingaben taetigen. Diese Funktion bietet auch das Hauptprogramm MB, wo der Sysop ohne eingeloggt zu sein, einem User zuschauen kann.

Der Befehl FORCE erlaubt es, einen String als Eingabe auf einen anderen Port zu schicken.

Der Befehl MESSAGE ermoeglicht es, eine Nachricht einen anderem Port auf den Bildschirm zu schicken.

Damit sich User komfortabel unterhalten koennen, bietet das AMMS zwei Chat-Systeme. Ein Fullscreenchat (FCHAT) und ein Zeilenchat (XCHAT oder WCHAT). Fuer den Zeilenchat existiert inzwischen auch ein Chatlink, der es ermoeglicht, dass mehrere Mailboxen gleichzeitig den Zeilenchat benutzen koennen und somit einen grossen Chatverbund bilden.

Das Message-, PM- und Filesystem sind fest im AMMS eingebaut.

Hinweise zum Message, PM- und File-System stehen weiter unten im Text.

Genauere Informationen stehen in den Textfiles MSG.DOK, FILE.DOK, NETZWERKE.DOK und FASTNET.DOK.

Das AMMS bietet neben dem PM-System auch ein ONLINE-Message-System.

Online-Messages sind kurze Texte, die sich die User gegenseitig schicken koennen. Dabei verschickt am Anfang ein User eine Online-Message mit dem Befehl PMSG.

Befindet sich der Zieluser im System, so wird ihm die Nachricht angezeigt, sobald er sich in der Befehlsebene oder im Menue aufhaelt. Er antwortet sofort ohne den Befehl PMSG extra aufzurufen.

Befindet sich der User gerade nicht im System, so wird die Nachricht im User-Directory (MBUDIR:<USERNAME>/PMSGS/) gespeichert und beim naechsten Login angezeigt.

Speicherverbrauch :

Das Hauptprogramm belegt ca. 800 KByte (bei 200 Usern und 400 Protokoll-eintraegen), wobei sich dieser Speicherbedarf mit der maximalen Useranzahl, der Protokollgroesse und der Anzahl der residenten Befehle veraendern kann. Die Ports belegen ca. 300 KByte, wobei Consolen ca. 50 KByte mehr belegen. Die ausfuehrbaren Befehle (mbcom:) sollten aus Geschwindigkeitsgruenden in einer RAM-Disk stehen und verbrauchen dann zusaetzlich ca. 200 KByte. Eine kleine Mailbox mit einem Port und einer Consolen sollte daher nicht auf einem Rechner mit weniger als 2.0 MByte laufen. Der Speicheraufwand ist bei einer hoeheren Portanzahl vergleichsweise gering, weil AMMS nicht mehrfach das Hauptprogramm in den Speicher laedt. Eine 20-Port-Box (1000 User/ 2000 Protokolleintraege) laeuft schon ab 8 MByte Fastram.

Messagesystem :

Das Messagesystem ist in einer sehr komplexen Form geschrieben. Die einzelnen Messages werden nicht wie ueblich in einzelnen Files gespeichert, sondern in nur zwei Verwaltungsfiles pro Brett mit maximal 800 Messages verwaltet. Durch diese Verwaltungsart ist eine sehr effektive und komfortable Leseumgebung moeglich, was der READ-Befehl mit ueber 50 Funktionen beweist. Die eigentlichen Vorteile bei dieser Verwaltungsart sind, dass alle Messages sehr schnell erreichbar sind und dass ca. 300 KByte pro Brett gespart werden, was bei 100 Brettern mit je 800 Messages schon ca. 30 MByte ausmacht. Die maximale Brettanzahl ist auf 1600 begrenzt, was insgesamt bei je 800 Msgs pro Brett eine Summe von 1.280.000 Messages (ca. 2,0 (19,7) GByte Platten-speicher bei 1.5 (16) KByte durchschnittliche Messagelaenge) ergibt. Die Beschraenkung 1600 wird ab der Version 2.1 auf ca. 5000 erhoehrt.

Damit der User nicht immer alle Bretter angezeigt bekommt, besitzt er eine Brettliste, in der er die Bretter eintraegt, die er lesen moechte. Diese Brettliste wird vom Befehl RN (READ/NEW) benutzt, um die neusten Messages aus den gewuenschten Brettern anzuzeigen. Ebenso wird diese Brettliste von den Zerberus- und Fastnet-Points benutzt.

Alle User sind mit ihrem Eintrag sofort ein ZCONNECT-Point (inc. MAPS). Das kann deshalb so realisiert werden, da fuer die User-Points kein zusaetzlicher Speicher benoetigt wird. Die Messages werden beim Pointen online aus den Brettern exportiert und gepackt. Optional kann man sich aber die Daten vorpacken bzw. vorpacken lassen.

Damit die schon gelesenen Messages nicht nochmal als Neue angezeigt werden, wird fuer jedes Brett die hoechste Messagenummer der schon gelesenen Messages gespeichert. Dabei fuehren die Messages eine fortlaufende Nummer mit sich, die genau die zeitliche Reihenfolge der Speicherung der Messages darstellt.

Weitere Informationen zum Message-System stehen im @("Message-Guide" link amms:dok ← /msg.guide).

PM-System :

Das PM-System baut auf dem Messagesystem auf und unterstuetzt maximal 800 private Messages pro User. Die Messages sind in den beiden Verwaltungsfiles "PM" und "PM-Tab" gespeichert, die in den User-Directories "mbudir:<username>/" stehen.

Der normale User sollte nicht max. 800 PMs zur Verfuegung gestellt bekommen, weil sonst eine sehr hohe Plattenkapazitaet notwendig ist. Zu empfehlen ist eine PM-Anzahl zwischen 20 und 50. Sie kann mit dem Befehl EDIT USER geaendert werden.

Mit dem Befehlen MAIL und SEND koennen PMs gelesen und versendet werden.

Alle PMs, die an ALL geschickt werden, werden in das Brett :SYSTEM/PMALL gespeichert. Auf dieses Brett darf kein User LESE- oder SCHREIB-Zugriff (----) haben, da sonst PMs an ALL nicht korrekt importiert werden koennten. Das PM-System schaut beim Aufruf des Befehls MAIL nach, ob neue PMs an ALL existieren. Wenn ja, dann werden die neuen PMs importiert. Die Abfrage, ob

eine PM neu oder alt ist, verläuft wie beim RN-Befehl über die höchste gelesene Messagenummer. Sollte diese Messagenummer beim Brett PMALL für alle User zurückgesetzt werden, so werden alle PMs an ALL nochmal neu importiert. Will ein Sysop eine PM an ALL löschen, so muss er mit Hilfe des READ-Befehls das Brett :SYSTEM/PMALL auswählen und die entsprechende PM löschen.

Filesystem :

Das Filesystem besitzt pro Filebrett ein Verwaltungsfile und ein Directory, worin alle Files des entsprechenden Bretts stehen. Pro Brett können bis zu 1400 Files mit vielen Zusatzinformationen verwaltet werden. Momentan können max. 640 Filebretter eingerichtet werden, zur Version 2.1 wird diese Anzahl deutlich erhöht.

Der Befehl FILES zeigt brettweise die Files in einem sehr komfortablen File-Requester an. Dabei können die Files mit Hilfe der Cursortasten ausgewählt und viele Funktionen auf sie angewendet werden (z.B. Archiv anschauen, Archiv überprüfen, Textfile anschauen, Bild über die Console anschauen, File löschen, File downloaden usw.). Mit über 50 Funktionen und mehreren selbst definierbaren Funktionen ist der Files-Befehl ein sehr mächtiges Instrument.

Alle Files können zum Download markiert werden, wobei die Markierungen nicht beim Verlassen des Systems verloren gehen. Sie werden im User-Directory unter dem Namen "DOWNTAB.DAT" abgespeichert. Der Batchdownload ist wie der Batchupload brettweise oder brettübergreifend möglich.

Damit der User Filebretter selektieren kann, besitzt er wie beim Messagesystem eine Filebrettliste. In diese Liste trägt der User alle Bretter ein, die er angezeigt bekommen möchte. Mit dem Befehl FN (FILES/NEW) werden nur die neuen Files der Bretter aus der Filebrettliste angezeigt. Mit dem Befehl FILES werden alle Files der Bretter aus der Filebrettliste angezeigt.

Damit die schon angezeigten Files nicht nochmal als neue Files angezeigt werden, wird für jedes Brett die höchste Filenummer der schon angezeigten Files gespeichert. Dabei führen die Files eine fortlaufende Nummer mit sich, die genau die zeitliche Reihenfolge der Importierung der Files in das Brett angibt.

Weitere Informationen zum File-System stehen im @("File-Guide" link amms:dok/file. ← guide) und im Helpfile des Befehls FILES (HELP FILES).

File-PM-System :

Das File-PM-System basiert auf dem Filesystem. Dabei wird die Möglichkeit des Versendens privater Files in einem Brett genutzt. Das Filebrett :SYSTEM/PRIVAT wird zum Privatfilebrett ernannt. Wird ein Privatfile gesendet, so muss mind. ein Zieluser angegeben werden. Nur die Sysops, der Sender und die Zieluser haben Download- und Lösch-Zugriff auf das File. Für die anderen User existiert das File nicht.

Alle User, die Privatfiles versenden duerfen, muessen Upload-, Download- und Loeschzugriff (-UDE) auf das Filebrett :FILES/PRIVAT/PRIVAT haben.

Der Batchbefehl FMAIL schaut nach, ob ein Privatfile vorliegt und geht gegebenenfalls in den FILES-Befehl (entspricht vergleichbar dem MAIL). Mit Hilfe des SMAIL-Befehls kommt man immer in das Privatfilebrett und kann Privatfiles uploaden (entspricht vergleichbar dem SEND). Die User koennen das Brett auch in ihre Filebrettliste aufnehmen oder direkt anwaehlen.

Disk Tool :

Sollte die Filebrettanzahl nicht reichen oder CDs eingebunden werden, so kann auf den Befehl DISK TOOL zurueckgegriffen werden. Er ist aehnlich dem Befehl Files aufgebaut und ist fuer die Einbindung von CDs sehr geeignet. Genauere Informationen stehen im Hilfstext zum Befehl DISK TOOL (AMMS> help disk tool).

Verwaltung der File- bzw. Messagebretter :

Die Verwaltung der File- und Msgbretter ist nach dem gleichen Prinzip aufgebaut:

Die Zugriffe S=Sysop, R=Read, W=Write und D=eigene Message loeschen oder aendern bzw. S=Sysop, D=Download, U=Upload und E=eigenes File loeschen oder Files anschauen werden fuer jedes Brett fuer jeden einzelnen User bestimmt. Dafuer existieren die Befehle EDIT BOARD bzw. EDIT FBOARD, womit alle Zugriffe der User auf ein Brett gesetzt werden und EDIT ACCESS bzw. EDIT FACCESS, womit alle Zugriffe eines Users auf alle Bretter gesetzt werden.

Zusaetzlich zu der Vergabe von Rechten gibt es noch ein Brettsysop, der alle Rechte auf sein Brett besitzt. Dieser wird beim Erzeugen des Bretts (MAKE BOARD bzw. MAKE FBOARD) angegeben und kann durch die Befehle INFO BOARD und INFO FBOARD angezeigt werden. Er ist der eigentliche Verwalter des Bretts und sollte fuer den reibungslosen Betrieb verantwortlich sein. Neben dem Brettsysop koennen noch andere User den Sysop-Zugriff bekommen. Diese User haben fast die gleichen Moeglichkeiten wie der Brettsysop bei der Verwaltung des Bretts.

Automatisches Eintragen neuer User und deren Zugriffe :

Wenn sich ein neuer User mit Hilfe des APPLICATION-Befehls eintraegt, so werden alle Brettzugriffe und alle anderen Userdaten, die Zugriffe bestimmen, vom Systemuser NEWUSER als Vorgabe kopiert. Somit ist der neue User mit den Zugriffen des User NEWUSER im System eingetragen. Ebenso bekommt er das komplette User-Directory des Users NEWUSER kopiert. Er kann sich sofort nach dem Antrag unter seinem Usernamen einloggen. Der Eintragsvorgang wird im Textfile MBDAT:ANTRAEGE.DAT mitprotokolliert und kann ueber den Befehl SHOW APPLICATIONS angeschaut werden.

Ist die maximale Useranzahl im System erreicht, so kann sich kein neuer User eintragen.

Vereinfachte Verwaltung von Zugriffen ueber spezielle Systemuser :

 Wie schon beim Message-System erwaehnt, kann fuer jeden User der Zugriffsmodus in jedem Brett einzeln bestimmt werden. Diese Verwaltung laesst alle Moeglichkeiten der Zugriffsvergabe zu, ist aber bei Levelorientierten Zugriffsvergaben sehr arbeitsintensiv, wenn der Sysop fuer jeden User einzeln die Zugriffe setzen muss.

Deshalb kann diese Art von Zugriffsvergabe wesentlich vereinfacht werden, indem man Systemuser einrichtet, die die entsprechenden Zugriffe fuer einen Level beinhalten. Soll nun ein User einen hoeheren Level und damit auch mehr Zugriffe bekommen, so geht man in den Befehl EDIT USER und waehlt den User an. In der obersten Zeile, wo man zwischen ENDE und SPEICHERN UND ENDE waehlen kann, gibt es auch noch die Moeglichkeit EINSTELLUNGEN VON ANDEREM USER HOLEN. Wenn man diesen Punkt anwaehlt und Return drueckt, muss ein Username angegeben werden. Dieser Username muss nun der Systemuser sein, der die entsprechenden Zugriffe fuer den User enthaelt. Nun werden alle Brettzugriffe und Zugriffe vom Systemuser zum User kopiert, und wenn man nun die Userdaten vom User speichert, hat der User nun den Zugriff, die der Sysop dem Level zugeordnet hat. Ein Systemuser ist ein normaler User mit dem Schutz-Flag y (sYstemuser). Die Flags fuer Protokoll, Status und Login sperren und Systemuser werden beim Kopiervorgang nicht mitkopiert.

Verwaltung der Zugriffe auf Befehle :

 Befehle sind die Schnittstelle zwischen User und Betriebssystem des AMMS. Da man fast alles mit den Befehlen des AMMS tun kann, darf nicht jeder User auf alle Befehle Zugriff haben. Dazu existiert eine etwas komplizierte, aber sehr maechtige Befehlsverwaltung.

Alle User besitzen einen Level von 0 - 9999 und 16 Zugriffsbits. Der Level dient der eigentlichen Zugriffsvergabe, aber ueber die 16 Zugriffsbits koennen User zeitweise Rechte auf hoeher privilegierte Befehle bekommen ohne den entsprechenden Mindestlevel zu besitzen.

Befehle haben folgende Level zur Zugriffsvergabe zur Verfuegung :

Userlevel = Mindestlevel, damit User Zugriff auf Befehl bekommt
 Sysoplevel = Mindestlevel, damit User Sysopzugriff auf Befehl bekommt
 Batchlevel = Der User bekommt innerhalb eines Batchbefehls diesen Level zugeordnet, wenn sein eigener Level kleiner ist. Hiermit kann einem User innerhalb eines Batchbefehls vorruebergehend ein hoeherer Level gegeben werden, damit er definierten Zugriff auf privilegierte Befehle bekommt.

Beispiel :

Der Befehl INFO SYSTEM benoetigt zur Anzeige des Plattenspeichers den DOS-Befehl INFO. Dieser ist aber nur ueber den Befehl EXTERN zu erreichen, der ueber einen Handler vollstaendigen Amiga-Dos-Zugriff bietet. Innerhalb des Batch-

befehls kann nun dem User definierten Zugriff auf solch einen Befehl gegeben werden.

Befehle haben folgende Zugriffsbitreihen zur Verfuegung :

Useracc = Wenn der User mindestens ein Zugriffsbit besitzt, das in dieser Bitreihe gesetzt ist, so hat er Zugriff auf den Befehl, auch wenn er einen zu niedrigen Level besitzt.

Sysopacc= Wenn der User mindestens ein Zugriffsbit besitzt, das in dieser Bitreihe gesetzt ist, so hat er Sysopzugriff auf den Befehl, auch wenn er einen zu niedrigen Level besitzt.

Batchacc= Der User bekommt innerhalb eines Batchbefehls die Zugriffsbits zugeordnet. Dabei behaelt er alle seine schon vorhandenen Zugriffsbits und kann somit nur mehr Bits erhalten. Wie beim Batchlevel hat es den Sinn, einem User kurzzeitig mehr Zugriffsrechte zu geben als er eigentlich besitzt.

Beispiel:

Im Menu muessen staendig Texte von Disk geladen werden. Diese Texte koennen aber nur durch hoeher privilegierte Befehle oder Befehlsfunktionen geladen werden, weil die Angabe eines Filenamens noetig ist. Damit der User dennoch definierten Zugriff darauf bekommt, wird ein Bit (in den vorgegebenen Menues Bit 6) gesetzt, das der User normalerweise nicht besitzt. Dieses Bit ist ebenfalls bei den entsprechenden Befehlen bzw. Befehlsfunktionen im Useracc gesetzt. Somit hat der User innerhalb des Menues auf Befehle Zugriff, die er normalerweise nicht benutzen kann.

Durch ein geschicktes Spiel aus Level und Zugriffsbits koennen sehr einfach zusaetzliche Befehle erstellt und ein komplexes Menuesystem aufgebaut werden.

Befehlsaufrufe :

Es gibt grundsaeztlich zwei Arten von Befehlen. Befehle mit einem Befehlswort (z.B. Dir und CD) und Befehle mit zwei Befehlsworten (z.B. Show Command und Show User), wobei die Befehlsworte mit mindestens einem Space getrennt sein muessen. Alle Befehlsnamen koennen abgekuerzt werden. Die Abkuerzungen bestimmt derjenige, der ein Befehl mit Hilfe des Befehls EDIT COMMAND/ADD eintraegt. Der User muss mindestens soviele Buchstaben eingeben, wie der Befehlsname am Anfang Grossbuchstaben besitzt. Dies gilt fuer jedes Befehlswort. Damit nicht versehentlich Befehle eingetragen werden, die die gleiche Abkuerzung besitzen, wird beim Eintragen die Abkuerzung auf Eindeutigkeit im System ueberprueft.

Damit ein Befehl mehrere Aufgaben uebernehmen kann, gibt es Optionen. Diese Optionen muessen direkt hinter dem letzten Befehlswort stehen und beginnen immer mit einem Slash (/). Die Optionen werden ohne Slash im Befehl EDIT COMMAND eingetragen, und ihnen ist ebenfalls ein Zugriffslevel und eine Zugriffsbitreihe zugeordnet, so dass der Zugriff auf Optionen genau definiert werden kann.

Auch Optionen koennen nach dem gleichen Prinzip, wie bei den Befehlsworten, abgekuerzt werden. Die maximale Optionsanzahl ist auf 6 beschraenkt. Batchbefehle koennen auch Optionen auswerten, in den Variablen O1 bis O6 bzw.

OS fuer Sysopzugriff steht entweder ein "T" fuer Option gesetzt oder ein "F" fuer Option nicht gesetzt. Diese Variablen koennen in den Batchbefehlen unter anderem mit Hilfe des IF-Befehls ausgewertet werden.

Damit ein Befehl von alleine eine Option benutzt, gibt es fuer jede Option ein Flag, das definiert, ob die Option beim Start an oder aus ist.

Die Benutzung einer Option bewirkt ein Invertieren dieses akt. Status an/aus.

Zu allen Befehlen steht ein Hilfstext zur Verfuegung, der ihre Funktion und die Funktionen der Optionen erkluert (HELP <Befehlsname>).